Vincent Kríž, Barbora Hladká

RExtractor: a Robust Information Extractor

Aplikace NLP, 10.5.2016
MFF UK

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic
Motivation

• large collections of documents
• efficient browsing & querying
• typical approaches
  – full-text search
  – metadata search

• semantic interpretation of documents → suitable DB & query language → user-friendly browsing & querying
Scenario

- **Cooperation between**
  - Information Extraction
  - Semantic Web
Scenario

- **Extracting knowledge base**
  - set of entities and relations between them
  - linguistic analysis (RExtractor)

- **Knowledge base representation**
  - Linked Data Principles
  - Resource Description Framework (RDF)
Scenario

• **Extracting knowledge base**
  - set of entities and relations between them
  - linguistic analysis (RExtractor)

• **Knowledge base representation**
  - Linked Data Principles
  - Resource Description Framework (RDF)
RExtractor Architecture

Conversion Component
- HTML
- TXT
- PDF

NLP Component
- Segmentation
- Tokenization
- Tagger
- Morphology
- Parser

Relation Extraction
- PML-TQ
- Database of queries

Entity detection
- PML-TQ
- Database of entities
RExtractor Architecture

Conversion Component
- HTML
- TXT
- PDF

NLP Component
- Segmentation
- Tokenization
- Tagger
- Morphology
- Parser

Relation Extraction
- PML-TQ
- Database of queries

Entity detection
- PML-TQ
- Database of entities
Conversion Component

- converts various input formats into a unified representation (XML)
NLP Component

• **Prague Dependency Treebank** framework
  - [http://ufal.mff.cuni.cz/pdt3.0](http://ufal.mff.cuni.cz/pdt3.0)

• **Tools**
  - segmentation & tokenization
  - lemmatization & morphology
  - syntactic parsing
(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations
Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations.
RExtractor Architecture
Entity Detection Component

- **Database of Entities**
  - entities specified by domain experts

  - tree queries better than regular expressions
    - coordination
    - several word forms in inflective languages
  - find the entity *current tangible assets* in the text *current tangible and intangible assets*
• find the entity *current tangible assets* in the text *current tangible and intangible assets*
(3) Accounting units, which keep books in simplified extent, create fixed items and reserves according to special legal regulations.
(3) **Accounting units**, which keep books in simplified extent, **create fixed items and reserves** according to special legal regulations.
Relation Extraction Component

- **Database of Queries**
  - queries formulated by domain experts
  - their formulation in the form of PML-TQ queries on dependency trees

- **RDF ready output**
  - triples \((subject, predicate, object)\)
  - each position
    - is annotated in a text \((text\ chunk)\)
    - has a specific **ontological concept** \((RDF Class)\)
Relation Extraction Component

- Accounting units' obligations

<table>
<thead>
<tr>
<th>Subject</th>
<th>Predicate</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity</td>
<td>hasToCreate</td>
<td>Something</td>
</tr>
<tr>
<td>Accounting units</td>
<td>create</td>
<td>fixed items</td>
</tr>
<tr>
<td>Accounting units</td>
<td>create</td>
<td>reserves</td>
</tr>
</tbody>
</table>
Case study on legislative domain

Legal texts

- specialized texts operating in legal settings
- they should transmit legal norms to their recipients
- they need to be clear, explicit and precise

Sentences

- simple sentences are very rare
- usually long and very complex

Legal texts are “generally considered very difficult to read and understand”. (Tiersma, 2010)
RExtractor Architecture

Conversion Component

- HTML
- TXT
- PDF

NLP Component

- Segmentation
- Tokenization
- Tagger
- Morphology
- Parser

Relation Extraction

- PML-TQ
- Database of queries

Entity detection

- PML-TQ
- Database of entities
RExtractor Architecture
RExtractor Architecture

Conversion Component
- HTML
- TXT
- PDF

NLP Component
- Segmentation
- Tokenization
- Tagger
- Morphology
- Parser

Relation Extraction
- PML-TQ
- Database of queries

Entity detection
- PML-TQ
- Database of entities
Automatic parsers for Czech

- trained on *newspaper texts*
- verification whether we can use the parser trained on newspaper texts or some modifications are needed
- **MST parser**
NLP Component

Czech Legal Texts Treebank (CLLT)

- Accounting Act (563/1991 Coll.)
- Decree on Double-entry Accounting for undertakers (500/2002 Coll.)
- automatically parsed, then manually checked
  - 1,133 manually annotated dependency trees
  - 35,085 tokens
Accounting units create fixed items according to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69
NLP Component

Re-tokenization

Accounting units create fixed items according to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69
Accounting units create fixed items according to § 16, 26, 31, 55 and 57 and not apply § 27, § 14, 39, § 51 - 55, § 58, 60 and 69
**RExtractor Architecture**

The RExtractor architecture is divided into Conversion Component and NLP Component. The Conversion Component takes HTML, TXT, and PDF inputs and converts them into XML. The NLP Component comprises Segmentation, Tokenization, Tagger, Morphology, and Parser. The Relation Extraction part uses PML-TQ and the database of queries. The Entity detection part also uses PML-TQ and the database of entities.
Entity Detection Component

Entities in CLTT

- Accounting subdomain
- Entities manually annotated by domain experts
  - Decree on Double-entry Accounting for undertakers (500/2002 Coll.)

Sample

(1) Vyhláška se vztahuje na účetní jednotky podle § 1 odst. 2 písm. a) a b) zákona, s výjimkou účetních jednotek uvedených v odstavci 2, a na účetní jednotky podle § 1 odst. 2 písm. d) až h) zákona.

(2) Z účetních jednotek uvedených v odstavci 1 se tato vyhláška nevztahuje na účetní jednotky podle § 19a zákona, pokud zvláštní právní předpis 1c) nestanoví jinak, a na účetní jednotky, jejichž účetnictví upravuje zvláštní právní předpis 1d). Dále se tato vyhláška, s výjimkou § 62 odst. 2 až 5, nevztahuje na účetní jednotky podle § 23a zákona.
Entity Detection Component

Initializing DBE with entities from CLTT

- Each (unique) entity parsed automatically by MST
- Automatic procedure takes an entity dependency tree and creates a PML-TQ query
Entity Detection Component

Experiment

- identify entities in the gold standard trees in CLTT
  - with re-tokenized tokens and *(very)* long sentences
- identify entities in the trees created by MST
  - with re-tokenized tokens and split sentences

<table>
<thead>
<tr>
<th>Parsing method</th>
<th>Extracted</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>16428</td>
<td>9549</td>
<td>6879</td>
<td>628</td>
<td>58.1</td>
<td>93.8</td>
</tr>
<tr>
<td>Automatic</td>
<td>16160</td>
<td>9278</td>
<td>6882</td>
<td>838</td>
<td>57.4</td>
<td>91.7</td>
</tr>
</tbody>
</table>

Results

- high False positives
- automatic parser has low influence on detection
RExtractor Architecture

Conversion Component
- HTML
- TXT
- PDF

XML

NLP Component
- Segmentation
- Tokenization
- Tagger
- Morphology
- Parser

Relation Extraction
- PML-TQ
- Database of queries

Entity detection
- PML-TQ
- Database of entities
Relation Extraction Component

Manual design of queries

- **Strategy**: cover maximum of relations with minimum of queries
- tree query expert
  - observes typical constructions for a given relation type
  - designs a query for the most frequent construction
  - goes through matches and redesigns the query if needed
Relation Extraction Component

Types of relations

- **Definitions**  
  • entities are defined or explained

- **Obligations**  
  • an entity is obligated to do something

- **Rights**  
  • an entity has right to do something
Query design & evaluation on CLTT

- Query design
  - on Accounting Act (563/1991 Coll.)
  - 5 queries for Definitions
  - 4 queries for Rights
  - 2 queries for Obligations

- Evaluation
  - on Decree on Double-entry Accounting for undertakers (500/2002 Coll.)
## Relation Extraction Component

### Results

<table>
<thead>
<tr>
<th></th>
<th>D</th>
<th>R</th>
<th>O</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td># of queries</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>Goldstandard</td>
<td>97</td>
<td>308</td>
<td>62</td>
<td>467</td>
</tr>
<tr>
<td>Extracted</td>
<td>70</td>
<td>255</td>
<td>41</td>
<td>366</td>
</tr>
<tr>
<td>True positive</td>
<td>53</td>
<td>206</td>
<td>36</td>
<td>295</td>
</tr>
<tr>
<td>False negative</td>
<td>44</td>
<td>102</td>
<td>26</td>
<td>172</td>
</tr>
<tr>
<td>False positive</td>
<td>17</td>
<td>49</td>
<td>5</td>
<td>71</td>
</tr>
<tr>
<td><strong>Precision (%)</strong></td>
<td>75.7</td>
<td>80.8</td>
<td>87.8</td>
<td>80.6</td>
</tr>
<tr>
<td><strong>Recall (%)</strong></td>
<td>54.6</td>
<td>66.9</td>
<td>58.1</td>
<td>63.2</td>
</tr>
</tbody>
</table>
Relation Extraction Component

Error analysis

<table>
<thead>
<tr>
<th>Error</th>
<th># of errors</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td>145</td>
<td>59.7%</td>
</tr>
<tr>
<td>Query</td>
<td>93</td>
<td>38.3%</td>
</tr>
<tr>
<td>Entity</td>
<td>5</td>
<td>2.1%</td>
</tr>
</tbody>
</table>

Results

- errors in automatic parsing
- query design
Scenario

- Extracting knowledge base
  - set of entities and relations between them
  - linguistic analysis (RExtractor)

- Knowledge base representation
  - Linked Data Principles
  - Resource Description Framework (RDF)
Legal ontologies

• **Document structure**
  - Act » Expression » Section

• **Document semantic**
  - Legal Concepts Ontology (`lexc:`)
    • represents entities & relations
    • Concept » Concept Version
    • Property » hasDefinition, hasRight, hasObligation, ...
  - Linguistic Ontology (`lingv:`)
    • links entities with their appearance in texts
Conclusion

• **general pipeline for extraction and representation** of information that is presented in raw texts
  
  - processes input texts by linguistically-aware tools
  
  - extracts entities and relations from sentence syntactic representation
  
  - Linked Data principles

• **Legal documents** as a pilot domain